Big machines should make a big difference in the training of tomorrow's scientists and engineers.

FROM SLIDE RULES TO SUPERCOMPUTERS

by Albert M. Erisman and Horst D. Simon

The contribution of supercomputers has been measured and accepted in a few well-publicized quarters, notably aerospace engineering, petroleum reservoir simulation, and weather forecasting. For the most part, however, this technology has been reserved for those few organizations and companies with the means to absorb a multimillion-dollar outlay. The result: supercomputers do not yet play a major role in the curricula of science and engineering schools in the United States, or anywhere else.

This scenario is changing, however. Through National Science Foundation (NSF) and state initiatives, the research community's access to supercomputers has expanded enormously in the past several years and is likely to continue to grow. Even so, the commonly shared requirements of supercomputer users across application disciplines tend to drain available resources. This is certainly the case with supercomputing expertise, owing to the underdevelopment of the educational infrastructure for generating talent.

Scientific computing, as well as the applications of supercomputers, is demonstrably different from computer science. Few schools tackle scientific computing. This article will explain why scientific supercomputing complements, but cannot become a part of, standard computer science disciplines.

The supercomputer clearly deserves more practical attention and a more precise training approach. To establish such an approach, we will review the current state of supercomputer education—what is being done to train scientists and engineers in the efficient use of this powerful technology. This discussion will conclude with a few thoughts on an educational imperative: how more researchers can come to supercomputing earlier in their careers. And, because this review seeks to challenge conventional thinking, it will finish with some pure conjecture on what impact increased use of this technology will have on scientific research.

Unlike any other type of computing device, the supercomputer radically changes a scientist's working perspective. Its extreme speed opens up new possibilities for exploration in large-scale analysis and simulation. Jack Worlton, a 30-year veteran of large-scale scientific computing with the Los Alamos National Laboratory in New Mexico, has compared the emergence of supercomputing to the introduction of the jet aircraft, which not only made travel faster, but spawned new businesses along with new methods of conducting business.

Applications involving supercomputers, whether scientific or commercial, tend to emerge slowly, however. By their very nature, supercomputers are the most unforgiving of computational devices. To put a slightly different twist on a rather dog-eared phrase, these devices do not suffer foolish mistakes lightly. Because of their computational power and advanced architecture, even small modifications in an application program can make a drastic difference in machine performance.

WRONG-HEADED APPROACH

Likewise, scientists and engineers who address the supercomputer as they would a traditional mainframe are apt to produce costly results. This type of wrongheaded approach—viewing a supercomputer as just a better mainframe—obscures the true potential of these machines and does injustice to their superior computational power.

Unfortunately, given the traditional approach to computing, it has been an easy mistake to make. In the past, applications scientists, applied mathematicians, numerical analysts, software specialists, and hardware designers have all contributed, in varying degrees, to the success of a given project. They made their respective contributions while working independently from each other.

For example, a scientist might develop a mathematical model, then pick up a numerical algorithm developed in isolation—or reinvented by the scientist. The numerical analyst could develop a solution algorithm without taking more than passing notice of the actual hardware on which the problem was to be solved. All this suited the hardware designers and computer architects well enough, because these people were busy building machines that were basically general purpose and could conveniently ignore the applications they were created to run.

The supercomputer has broken the barrier between these comfortably exclusive fields of expertise. While providing much greater computing power and memory, the new machine also introduces a different architecture and forces the scientist, software engineer, and mathematician all to reevaluate computational algorithms and even modeling assumptions. The functions of the three separate camps are linked. On vector computers, performance results demand that computational models be matched with computer architecture. This rethinking process promises to become even more important, as more parallelism in systems becomes a reality.

The interrelationship between applications, algorithm, and architecture holds, regardless of scientific discipline. For example, consider a common structural dynamics problem in structural engineering: in order to determine the response of a high-rise building to an earthquake or the behavior of a bridge to wind loading, structural engineers perform so-called vibration analysis. From a computational viewpoint, a vibration analysis consists of the solution of a large, generalized eigenvalue problem with sparse coefficient matrices. In short, this means structural engineers take a big, complicated structure and model it as a set

The supercomputer has broken the barrier between the comfortably exclusive fields of the scientist, mathematician, and engineer.

of discrete elements. Each element interacts with its neighbor, and the interaction is described by the masses and stiffnesses associated with each element. Knowing these relationships, engineers can use vibration analysis techniques to predict the behavior of a bridge, building, or other structure, and thus assure its integrity.

PROBLEM UPSIDE DOWN

The supercomputer has turned this typical engineering problem upside down. Before supercomputers, the engineer performed this analysis through a process called static condensation, a manual process which reduced the original detailed model to a coarser, smaller, manageable size. Why? Because modeling an entire structure was beyond the scope of available computers. Static condensation was tedious, time-consuming, and professionally demanding, adding to the design cost of a structure while detracting from the precision of its analysis.

The supercomputer has made it possible to solve an original, detailed vibration analysis problem directly, performing the mechanical aspects of the problem and reducing human error. But the job requires advanced algorithm work. Many structural engineers in the 1970s were content using outdated numerical methods, even if their mathematician colleagues considered them antiquated. Engineers didn't care; they were not trying to analyze larger models anyway.

With supercomputers, this view has changed. These machines can handle an analysis of structures with several thousands or 10,000 degrees of freedom, so that even moderate gains in performance due to algorithmic improvements can turn into significant economic gains.

For instance, a group of analysts recently added a state-of-the-art algorithm, the so-called block-shifted Lanczos algorithm, to MCS/NASTRAN, a well-known structural analysis package sold by MacNeal-Schwendler Corp. of Los Angeles. The use of this new algorithmic technique dramatically improved the capabilities of MSC/NASTRAN, both with respect to reliability and performance.

The achievement can be explained with a simple illustration: in the early 1970s, George Platzman, an oceanographer at the University of Chicago, developed a model of the Atlantic Ocean and attempted a vibration analysis in order to determine the normal modes of the ocean. In 1975, this problem was numerically unsolvable. Mathematical algorithms could not handle the difficulties of the problem, and the computers of 10 years ago were insufficient to achieve a brute force solution.

In the early 1980s, advances in algorithm work and the availability of a Cray-1 allowed researchers to produce a partial solution of the problem. Fifteen modes were computed in 2.72 seconds. The code failed, however, due to numerical difficulties. With the enhanced version of MSC/NASTRAN and a Cray X-MP, the entire model ran in only 8.2 seconds.

Advanced algorithms and computer hardware have, within a decade, transformed the Atlantic Ocean problem from an infeasible one into a trivial one. The importance of this example lies in the fact that this progress was not achieved in either discipline alone. Without the needs and requirements of the structural engineering community, numerical analysts would not have worked as intensively on the problem. Nor would they have achieved the necessary understanding of algorithmic assumptions inherent in the engineering problem. And only the availability of supercomputers, in turn, made it possible to consider such large-scale models. Hence, progress sprang from close interaction between researchers with differing academic backgrounds and professional goals.

What is the catalyst for this cooperative approach? The marketplace. The competitive pressures of doing business have forced companies to adopt this more enlightened arrangement. Unfortunately, similar cooperative approaches are not broadly filtering down to graduate departments throughout the country. As a result, opportunities to use supercomputers efficiently and creatively are being missed.

Scientific computing is a relatively mature phenomenon nevertheless, the offspring of scientists using computers, particularly supercomputers. As discussed earlier, its natural home lies at the intersection of engineering, science, mathematics, and computer science. Unfortunately, scientific computing remains something of an unclaimed curriculum on American campuses. Although not recognized as such, scientific computing is easily distinguished from computer science and deserves a separate educational approach.

Computer science is a well-defined academic field, traditionally concerned with hardware, languages, operating systems, software techniques, and theoretical issues in algorithms research. Other research topics such as numerical analysis, graphics, and parallel algorithms fall within the domain of computer science but, because these topics concentrate on tool building rather than basic research, they remain at the periphery. The situation is perhaps best illustrated by the position of FORTRAN in many computer science departments. FORTRAN is the basic programming language of engineering and scientific application programs, but is often considered an irrelevant topic in computer science education.

Researchers in application areas carry out scientific exploration using computers. Computer modelers are mainly concerned with experimental design, data collection, and the interpretation of results. Software issues and application-specific algorithms, together with data management and graphics, are subjects of recent and growing interest to these researchers.

Scientific computing draws its identity and its potential from the synergy of computer science and scientific exploration using computers. It furthers the cause of science, having become an accepted methodology for performing research in physics, chemistry, and certain types of engineering. As such, computer simulation is the research equivalent of theory and experimentation. Scientists use computer simulation to suggest new hypotheses and to support theory formulation. It provides an inexpensive way to verify theoretical results, substituting for elaborate experiments.

BIG BOXES TO BE SUPREME

Thus, scientific computing complements and augments theory and experimentation. In that the supercomputer has made computer simulation a valid and potent research alternative, the device is apt to become the most important research tool of the late twentieth century.

The movement toward supercomputing is now under way. Scientists and researchers realize that the supercomputer can produce better results, solve larger models, and achieve faster turnaround times than conventional mainframes. The truly knowledgeable practitioner of scientific computing also realizes that the switch from mainframe to supercomputer requires more than a simple code conversion.

To exploit the full power and receive the full benefit of supercomputing, the user must understand the underlying computer architecture, stay aware of new algorithm developments, and comprehend the computational structure of their own application code.

Why? New physical phenomena are being modeled constantly. This demands constant reassessment of modeling assumptions. Computational models being used today are often the by-product of a previous generation of computers. In computer simulation, the price of research success must be partially paid in increased researcher in-

Progress sprang from close interaction between researchers with differing academic backgrounds and professional goals.

volvement in solving computational questions. Only a disciplined, university-based approach to scientific computing will help fill the present-day gap between supercomputer technology and academic research.

Computer science departments, acting on their own, can't get the job done. Scientific computing education requires an interdisciplinary approach. Effective research groups will consist of scientists, engineers, and technicians together with computer scientists and applied mathematicians, trained in areas such as numerical analysis, probability theory, statistics, and software engineering.

A few schools have adopted this interdisciplinary approach, including Cornell University's Center for Theory and Simulation in Science and Engineering, and, perhaps the leader in this field, the University of Illinois' Interdisciplinary Research Center (IRC).

IRC is closely associated with two additional centers at the University of Illinois—the Center for Supercomputing Research and Development (CSRD) and the National Center for Supercomputing Applications (NCSA). CSRD is focused on computer science research; NCSA is devoted to scientific computing. IRC bridges the gap between the computational tool builder and tool user.

IRC provides a forum for students, faculty, and visitors to experiment and to exchange ideas about research and computational techniques. The goal is to eliminate common computational bottlenecks and improve code efficiency. Both the IRC and the Cornell center, operating under similar premises, are attempting to interrelate algorithm, architecture, and application. Their approach will soon yield rich rewards, both in scientific advances and the more efficient use of computer technology.

At the majority of American universities, however, scientific computing has yet to be established as a discipline in its own right. Most computer science departments have not encouraged scientists pursuing research in areas such as mathematical software and the implementation of numerical algorithms. While such research has great value to scientific computing, it remains at the periphery of the traditional computer science department.

Similarly, a physicist or chemist runs the risk of having research judged to be second rate if it concentrates in the area of computer implementation rather than theoretical investigation.

The solution? Improve the education of engineers and scientists by removing the research barriers and raising the academic standing of scientific computation. Create interdepartmental research groups and interdisciplinary programs in scientific computing. Promote the interaction of practitioners in both computer science and

CARTOON BY ARTEMAS COLE

application fields, while avoiding the urge to develop independent—and eventually isolated—scientific computing groups.

Incentives must foster this change. Federal government participation is crucial. Only federal funding can provide the necessary combination of monetary stimulus and national attention. NSF has taken the first steps in this direction. Recently, NSF took an extremely positive step by combining its Office of Advanced Scientific Computing with corresponding offices responsible for computer science and electrical engineering. The result is the new Directorate for Computing.

One hopes this action will be quickly followed by the next logical step: creation of a program that encourages universities to establish interdisciplinary research and teaching units in scientific computing. (This is one of the key suggestions of the Rheinboldt Report on Future Directions in Computational Mathematics, issued under NSF's sponsorship in 1985.)

Computer simulation is now a well-accepted scientific research alternative. At present, so-called advances in this field are most apt to involve improvements to the speed of computers. True progress in computer simulation lies elsewhere. Researchers not only need access to high-speed computational power, but also to expert help in the latest algorithm developments and data display. Computational science must be conducted in an environment that interrelates software, algorithms, graphics, networking, and hardware.

Current forecasts call for the supercomputers of the 1990s to operate at speeds 100 times faster than those in use today. What does this mean to scientific progress? Results are difficult to predict, precisely because progress is a product of human ingenuity and innovation. Providing faster and larger computers will lead to the same results, produced more efficiently and in dramatically less time.

True qualitative improvements in scientific research, however, will be obtained only when researchers, numerical analysts, and computer scientists collaborate to create a new working environment for exploiting the promise of the supercomputer. ⊛

Albert M. Erisman, who is director of the Engineering Technology Applications (ETA) Division of Boeing Computer Services, holds an undergraduate degree in mathematics from Northern Illinois University and both graduate and doctorate degrees in applied math from Iowa State University.

Horst D. Si
Computati
Boeing Co
ploma in m
nische Uni
Germany,
the Univer